CA 분석기의 오류진단과 오류가 있는 입력수열의 오류탐지

조성진*, 권민정**, 임지미**, 김진경**, 박영규**

Fault Diagnosis in the CA Analyzer and Fault Detection of the Input Sequence

Sung-Jin Cho* · Min-Jeong Kwon* · Ji-Mi Yim** · Jin-Gyoung Kim** · Young-Gyu Park**

이 논문은 2009년도 정부재원(교육과학기술부 인문사회연구역량강화사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2009-371-B00008).

요 약

본 논문에서는 셀룰라 오토마타를 이용하여 테스트되는 회로에 상관없이 마지막 테스트 압축치가 일정하게 되도록 조직함으로써 오류를 진단한다. 이 방법은 셀룰라 오토마타를 이용한 테스트 절차를 간결하고 명확하게 한다. 그리고 셀룰라 오토마타의 상태전이행렬의 역행렬을 사용하여 오류가 있는 입력수열의 오류를 탐지한다.

ABSTRACT

In this paper, we diagnose the fault in the CA analyzer by setting up the initial value such that the final test signature is a constant regardless of the circuit being tested. This method makes the CA test procedure short and clear. In addition, we detect the fault of the faulty input sequence by using the inverse matrix of the CA state transition matrix.

키워드

셀룰라 오토마타, 상태 전이 행렬, 입력 수열, 압축치, 원시 다항식

* 부경대학교 (교신저자)
** 부경대학교

접수일자 : 2009. 04. 07
심사완료일자 : 2009. 04. 21

Ⅰ. 서 론

CA는 Neumann [1]에 의해 스스로 조직화하고 재생산할 수 있는 모델로 처음 소개되었다. Wolfram [2]은 스스로 조직화 할 수 있는 시스템을 위한 수학적 모델로써 CA의 연구를 개척하였고 모든 셀이 선형으로 배열되어 있으며 각 셀이 0과 1, 두 상태를 가지고 다음 상태가 자기 자신과 인접한 두 이웃에 의하여 갱신되는 3-이웃 (three-neighborhood) CA를 제안하였다. 또한 Das 등은 행렬 대수를 이용하여 CA의 특성을 분석하는 방법을 도입하였다 [3, 4]. CA는 테스트 패턴 생성, 의사난수생성기, 오류정정부호기, 암호 등 많은 분야에 응용되고 있다 [3, 5-9].

두개의 상태와 세 개의 이웃을 가진 각 셀의 행동분석은 비교적 최근에 이뤄졌다 [10]. Group CA의 상태전이 행동 분석은 많은 연구원들에 의해 실행되었다 [3, 5, 11-13]. group CA의 상태전이 행렬은 정칙이며 정칙인 선형 기계의 연구는 많은 관심의 대상이다. Cho 등 [14, 15]과 다른 연구원들 [16, 17]이 정칙인 선형 기계를 연구했고 정칙인 CA의 여러 가지 성질들은 많은 곳에서 응용되고 있다 [4, 13, 17, 18].

나눗셈 다항식에 의한 self-testing은 McAnney 와 Savir [21]에 의해 연구되었다. 이 방법에 의해 회로를 구현하는 것은 복잡하기도 하지만 이 방법을 사용하기 위해 도입해야 하는 상반다항식을 구하는 것도 간단하지 않다. 나눗셈 다항식은 피드백 연결(feedback connection)에 의해 정의되는데 이것은 두 가지로 표현할 수 있다. 그 중 한 형태는 정확한 나머지를 만들어내지만 다른 하나는 항상 정확한 나머지를 만들어내는 것은 아니다. 그래서 McAnney와 Savir [21]는 각각의 경우로 나누어 나눗셈에 의한 self-testing을 증명했다 [21]. 본 논문에서는 group CA의 상태전이 행렬과 그것의 역행렬을 이용하여 오류를 진단함으로써 이런 어려움을 극복하고자 한다. [20]의 방법을 사용하면 크기가 큰 group CA의 상태전이 행렬도 쉽게 찾을 수 있다.

오류가 없는 입력수열에서 하나의 single test만 실패했다고 가정하자. LFSR을 사용하여 실패한 test를 확인하는 방법으로는 압축치 분석 다항식의 상반다항식을 사용하는 것이 있다. 상반다항식으로 만들어진 LFSR에 의해 생성된 수열은 나눗셈 다항식을 사용하여 LFSR로부터 얻은 수열과 정확히 역순이다. 그러나 group CA에서는 이 수열을 만들어내기 위해 상태전이 행렬의 역행렬을 사용하면 된다.

본 논문에서는 test의 마지막 압축치가 test되는 회로에 관계없이 상수가 되도록 초기값을 설정함으로써 CA 분석기의 오류를 진단하겠다. 이 방법은 CA test 과정을 간결하고 분명하게 만든다. 게다가 오류가 있는 입력수열의 오류는 CA 상태전이 행렬의 역행렬을 사용함으로써 탐지할 수 있다.

입력수열의 길이 m은 순차적인 n-셀 CA의 cycle 길이보다 작게 되도록 제한해야 한다. 만약 CA가 최대 길이를 가지면 2^n-1 개의 패턴 후에 그 cycle은 다시 반복되고, $m>2^n-1$이면 $100\cdots00$의 패턴이 여러 번 나타날 수 있다.

본 논문의 구성은 먼저 2장에서는 $GF(2)$ CA와 관련된 배경지식을 몇 가지 소개하고 3장에서는 test되는 회로와 상관없이 마지막 test 압축치를 상수로 만들기 위한 초기값을 선택하는 방법을 나타내겠다. 4장에서는 오류가 없는 회로 S의 압축치와 관측된 부정확한 압축치 S'를 사용하여 주어진 입력 수열의 오류 bit를 찾는다.

Ⅱ. CA 배경지식

CA는 시간과 공간이 이산적인 물리적 시스템의 수학적인 구현이며 0 또는 1의 값을 가질 수 있는 수많은 셀들로 구성되어 있다.

그 셀들은 이산적인 시간 단계에서 국소적인 이웃들에게만 의존하여 결정하는 규칙에 따라 갱신된다. 실제로 각 셀은 저장원소(D flip-flop)와 다음 상태전이 함수(next state function)를 구현하는 조합논리로 이루어져 있다. 가장 간단한 경우는 두개의 상태와 3-이웃을 가지는 1차원 CA이며 여기서 각 셀들은 1차원 내에서 선형적으로 배열되어 있다. 셀의 다음 상태는 그 자신과 그것의 두 이웃에 의해서만 결정된다고 가정한다 (3-neighborhood dependency).

CA를 정의하기 위해 아래의 표기가 사용된다 :

- i:일차원으로 배열되어 있는 각 셀들의 위치 ;
- t:시간 단계 ;
- $q_i(t)$:시간 t에서 i번째 셀의 상태 ;
- $q_i(t+1)$:시간 $(t+1)$의 i번째 셀의 상태.

3-이웃 CA의 다음 상태전이 함수(next-state transition)는 아래와 같이 표현될 수 있다:

$$q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$$

여기서 함수 f는 CA의 "rule"로 알려진 결합논리를 갖는 국소전이 함수이다.

두 개의 상태를 갖는 3-이웃 CA는 2^3개의 서로 다른 이웃 상태를 가질 수 있다. 그런 CA에 대해서 이런 이웃들의 형태와 다음 상태 사이에 총 2^{2^3}개의 서로 다른 대응이 존재할 수 있다.

셀의 다음 상태 함수가 아래의 표와 같은 형태로 표현된다면 그 산출물을 십진법으로 표현한 수를 그 셀의 rule number라 한다 [2].

이웃상태	111	110	101	100	011	010	001	000	rule
다음상태	0	1	0	1	1	0	1	0	90
다음상태	1	0	0	1	0	1	1	0	50

여기서 첫 행은 시간 t에서 인접한 세 개의 셀(i번째 셀의 왼쪽 이웃, i번째 셀 자신, i번째 셀의 오른쪽 이웃)의 가능한 8가지 상태를 배열한 것이다. 두 번째와 세 번째 행은 각각 시간 $t+1$에서 i번째 셀의 갱신된 상태를 나타낸다. 위의 rule에 대한 결합논리는 다음 식으로 표현될 수 있으며 여기서 $\oplus$는 XOR 논리를 나타낸다.

- rule 90 : $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$
- rule 150 : $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$

[정의 2.1] ([17]) CA 셀의 rule이 XOR 논리만 포함하고 있으면 linear rule이라 하고 XNOR를 포함하는 rule은 complemented rule이라 한다. 모든 셀들이 linear rule을 갖는 CA는 linear CA라 하고 반면에 XOR rule과 XNOR rule의 조합을 갖는 CA는 additive CA라 한다. AND-OR논리를 포함하는 rule은 nonadditive rule이라 한다.

[정의 2.2] ([17]) 1차원 CA의 모든 셀들이 같은 rule을 따르는 경우 그 CA는 uniform CA라 하고 그렇지 않으면 hybrid CA라 한다.

[정의 2.3] ([17]) CA의 제일 왼쪽과 오른쪽의 셀들이 0 상태와 연결되어 있으면 NBCA(null boundary CA), 양 끝의 셀들이 서로 연결되어 있으면 PBCA(periodic boundary CA), 첫 번째 셀의 왼쪽 이웃을 세 번째 셀로 정의하고 마지막 셀의 오른쪽 이웃을 마지막 셀로부터 두 번째 왼쪽 셀로 정의되면 IBCA(intermediate boundary CA)라 한다 (그림 2.1).

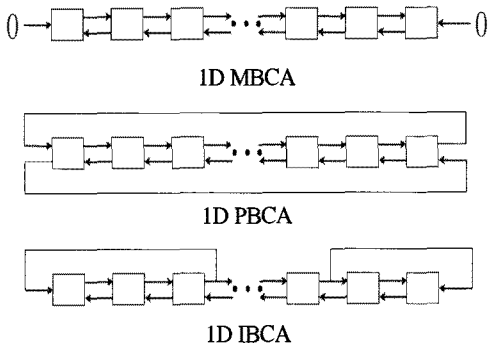

그림 2.1 Boundary에 따른 CA의 종류
Fig. 2.1 MBCA, PBCA and IBCA

이 논문에서는 특별히 언급하지 않은 CA는 모두 NBCA를 의미한다 [12].

n개의 셀로 이루어진 n-셀 CA는 $GF(2)$ 에서 연산이 이루어지는 $n \times n$ 상태전이 행렬로 나타낼 수 있다. 상태전이 행렬 $T = (t_{ij})$는 다음과 같이 만들어질 수 있다:

- $t_{ij} = 1$: i번째 셀의 다음 상태가 현재 j 번째 셀에 영향을 받는 경우
- $t_{ij} = 0$: 그 외의 경우

CA의 다음 상태는 현재 상태 벡터와 행렬의 곱으로 얻어진다. 만약 $f_t(x)$가 시간 t인 순간 CA의 상태를 나타낸다면, 시간 $t+1$ 순간의 상태와 $t+2$ 인 순간의 상태는 아래의 식으로 표현될 수 있다:

$$f_{t+1}(x) = T \cdot f_t(x)$$
$$f_{t+2}(x) = T \cdot f_{t+1}(x) = T^2 \cdot f_t(x)$$

같은 방법으로 p단계 후의 상태는

$$f_{t+p}(x) = T^p \cdot f_t(x)$$

이다.

[정의 2.4] ([22]) 인수분해되지 않는 n차 다항식 $p(x)$가 $x^m - 1$을 나눌 때, m의 최소값이 $2^n - 1$ 인 다항식을 n차 원시 다항식이라 한다.
최대길이를 갖는 CA(MLCA)라는 것과 그 상태전이 행렬의 특성다항식이 원시다항식이라는 것이 동치임은 잘 알려져 있다.

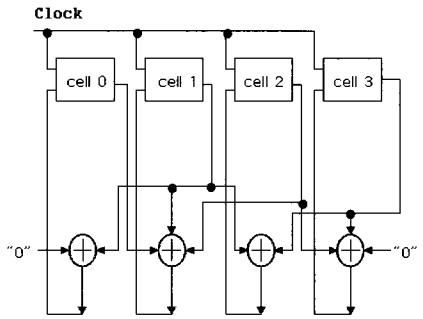

그림 2.2 Rule $<90,150,90,150>$를 갖는 최대길이 NBCA
Fig. 2.2 MLCA with rule $<90,150,90,150>$

[예제 2.5] Rule $<90,150,90,150>$를 갖는 4-셀 NBCA CA (그림 2.2)는 아래의 상태전이 행렬로 표현될 수 있다:

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

현재 상태 벡터가 $f_t(x) = [0\,1\,0\,1]$이면 그 다음 상태 벡터는

$$f_{t+1}(x) = T \cdot f_t(x) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

으로 얻을 수 있다. 이후부터는 역상태전이는 행렬론의 방법으로 구하겠다.

T의 특성다항식 $p(x)$는 $x^4 + x + 1$이고 이것은 원시다항식이므로 $<90,150,90,150>$는 MLCA이다. Group CA는 정칙인 행렬 T에 대응되므로 역행렬의 관점에서 CA의 특성화도 가능하다. Group CA에 대해서는 아래의 식을 만족하는 정수 k가 존재한다:

$$T^k = I \Leftrightarrow T^{-1} = T^{k-1}$$

앞으로 진행되는 상태전이가

$$f_{t+1}(x) = T \cdot f_t(x)$$

으로 표현된다면 역으로 진행되는 상태전이는

$$\begin{aligned} f_t(x) &= T^{-1} \cdot f_{t+1}(x) \\ &= T^{k-1} \cdot f_{t+1}(x) \end{aligned}$$

와 같이 표현된다.

III. CA 분석기의 오류진단

Self-test 절차는 압축치 분석기에서 초기값을 입력하고, pseudorandom test의 자극으로써 수열을 적용하고, 이미 계산되어져 있는 올바른 압축치와 압축치 분석기에서 남겨져 있는 마지막 압축치를 비교하는 과정으로 이루어진다.

다음의 정리에서는 테스트되는 회로에 관계없이 마지막 test 압축치가 모두 0이 되도록 초기값을 설정함으로써 test 절차를 보다 간결하게 할 수 있다.

[정리 3.1] n-cell group CA $\mathbb{C}$에 오류가 없다면 다음 과정에 의해 얻어진 압축치 s_f는 0이다.

Step 1. $\mathbb{C}$의 상태전이행렬을 T 두자.
Step 2. $\mathbb{C}$의 초기값을 0으로 두고, m-bit 수열을 $\mathbb{C}$에 입력하자. m cycle 후의 $\mathbb{C}$의 마지막 압축치를 s_1이라 두자.

Step 3. $T^{-m}s_1(:=s_2)$을 계산하자.

Step 4. $\mathbb{C}$의 초기값을 s_2로 두고, m-bit 수열을 $\mathbb{C}$에 입력하자. m cycle 후의 $\mathbb{C}$의 마지막 압축치를 s_f라 두자.

【증명】 $\mathbb{C}$의 초기값을 모두 0으로 두고, m-bit 수열 $Z=(z_0,z_1,\cdots,z_{m-1})$라 하자. Z의 첫 번째 bit z_0를 입력하고 난 후의 $\mathbb{C}$의 압축치는

$$T\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}$$

이다.

두 번째 bit z_1을 입력하고 난 후의 $\mathbb{C}$의 압축치는

$$T\left(T\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}=T^2\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}$$

이다.

같은 방법으로 m번째 bit z_{m-1}을 입력한 후의 $\mathbb{C}$의 압축치는

$$\begin{aligned}&T\left(T^{m-1}\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{m-2}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+T^{m-2}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+\cdots\right.\\&\qquad\left.\cdots+T\begin{bmatrix}z_{m-3}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}\\&=T^m\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{m-1}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+T^{m-2}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+\cdots\\&\qquad\cdots+T\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}=s_1\end{aligned}$$

이다.

step 3에 의하여 s_2는 아래와 같다.

$$\begin{aligned}s_2=T^{-m}s_1=&\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{-1}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+T^{-2}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+\cdots\\&\cdots+T^{-m+1}\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+T^{-m}\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}\end{aligned}$$

계속해서 $\mathbb{C}$의 초기값을 s_2라 두고, Z의 첫 번째 bit z_0를 입력하고 난 후의 $\mathbb{C}$의 압축치는

$$\begin{aligned}&T\left(\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{-1}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+T^{-2}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+\cdots\right.\\&\qquad\left.\cdots+T^{-m+1}\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+T^{-m}\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}\\&=T^{-1}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+T^{-2}\begin{bmatrix}z_2\\0\\\vdots\\0\end{bmatrix}+\cdots\\&\qquad\cdots+T^{-m+2}\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+T^{-m+1}\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}\end{aligned}$$

이다.

두 번째 bit z_1을 입력하고 난 후의 $\mathbb{C}$의 압축치는

$$\begin{aligned}&T\left(T^{-1}\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}+T^{-2}\begin{bmatrix}z_2\\0\\\vdots\\0\end{bmatrix}+\cdots\right.\\&\qquad\left.\cdots+T^{-m+2}\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+T^{-m+1}\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}\\&=T^{-1}\begin{bmatrix}z_2\\0\\\vdots\\0\end{bmatrix}+T^{-2}\begin{bmatrix}z_3\\0\\\vdots\\0\end{bmatrix}+\cdots\end{aligned}$$

$$\cdots + T^{-m+3}\begin{bmatrix} z_{m-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{-m+2}\begin{bmatrix} z_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

이다.

같은 방법으로 m번째 bit z_{m-1}을 입력한 후의 $\mathbb{C}$의 압축치는

$$T\left(T^{-m+(m-1)}\begin{bmatrix} z_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) + \begin{bmatrix} z_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = s_f$$

이다.

따라서 s_f는 0 벡터이다.

□

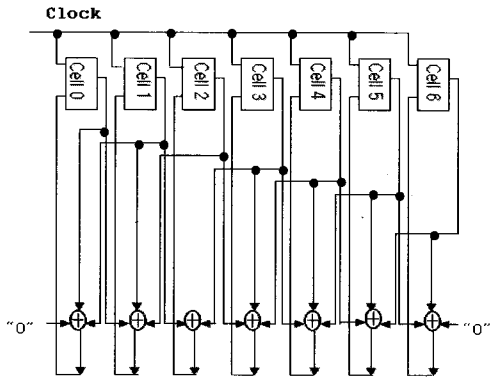

그림 3.1 $<150,150,90,150,150,150,150>$를 rule로 가지는 최대길이 Null Boundary CA
Fig. 3.1 MLCA with rule $<150,150,90,150,150,150,150>$

[예제 3.2] 원시특성다항식이 $x^7+x^5+x^3+x+1$인 7-cell 선형 group CA 를 $\mathbb{C}$라 하자. [20]$\mathbb{C}$의 전이규칙이 $<150,150,90,150,150,150,150>$임을 알 수 있다. 그러므로 $\mathbb{C}$의 상태전이 행렬 T는 아래와 같다.

$$T = \begin{bmatrix} 1100000 \\ 1110000 \\ 0101000 \\ 0011100 \\ 0001110 \\ 0000111 \\ 0000011 \end{bmatrix}$$

입력 수열을 $Z=(0,1,1,0,0,0,1,0,1,0,0,1,1,1,1)$라 하자. $\mathbb{C}$의 초기값을 모두 0으로 두고 수열 Z를 압축하자. Z의 15 bit가 모두 입력된 후에 압축치는 $s_1=(0,1,0,1,0,1,1)$이다.

s_1의 15번째 직전상태인 s_2를 구하는 과정에서 거슬러 가는 과정을 피하기 위해 간단한 접근방법인 T^{-1}를 사용한다. T^{-1}는 아래와 같다.

$$T^{-1} = \begin{bmatrix} 0111011 \\ 1111011 \\ 1100000 \\ 1101011 \\ 0000011 \\ 1101100 \\ 1101101 \end{bmatrix}$$

s_1을 초기값으로 하고 T^{-1}를 15번 적용하여 벡터 $s_2=(0,0,0,1,1,1,1)$를 얻을 수 있다.

마지막으로 s_2를 $\mathbb{C}$의 초기값으로 하고 수열 Z를 입력하면 마지막 상태 s_f는 $(0,0,0,0,0,0,0)$이다.

Ⅳ. 오류가 있는 입력수열의 오류 탐지

오류가 없는 수열에서 단지 하나의 bit에 오류가 생겼다고 가정하자. LFSR을 사용하여 오류 test를 식별하기 위해서는 압축치 분석기 다항식의 상반다항식을 사용하는 방법이 있다. 이 절에서는 오류 test를 식별하기 위해 group CA의 상태전이행렬의 역행렬을 사용하는 방법을 제안한다.

m-bit 수열 Z에서 하나의 bit가 오류일 경우에 오류 test를 식별하기 위해서 group CA의 상태전이행렬의 역행렬을 사용한다. 오류가 없는 수열 Z를 입력한 후의 $\mathbb{C}$의 압축치를 S라 하고, 하나의 bit가 오류인 수열 Z'을 입력한 후의 $\mathbb{C}$의 압축치를 S'라 하자. 우리는 $S \oplus S'$을 사용하여 Z'을 진단한다.

[정리 4.1] n-cell group CA $\mathbb{C}$의 초기값을 모두 0으로 두고, 오류가 없는 m-bit 수열 Z를 입력한 후의 압축치를 S라 하자. 그리고 하나의 bit에 오류가 있는 수열 Z'

을 입력한 후의 압축치를 S'이라 하자.($m < 2^n - 1$) 그러면 $S_e = T^k \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$를 만족하는 $k(0 \le k \le m-1)$가 존재한다.

(단, $S_e = S \oplus S'$, T는 $\mathbb{C}$의 상태전이행렬, Z'의 $(m-k)$번째 bit가 오류 bit이다.)

【증명】 n-cell group CA $\mathbb{C}$의 초기값을 모두 0으로 두고, 오류가 없는 m-bit 수열 $Z=(z_0,z_1,\cdots,z_{i-1},\cdots,z_{m-1})$를 입력한 후의 압축치를 S, 그리고 i번째 bit에 오류가 있는 m-bit 수열 $Z'=(z_0,z_1,\cdots,z_{i-1}{}^*,\cdots,z_{m-1})$을 입력한 후의 압축치를 S'라 하자. $\mathbb{C}$의 상태전이행렬을 T라 하고, $\mathbb{C}$의 초기값을 모두 0으로 두고 Z의 첫 번째 bit z_0를 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Z의 두 번째 bit z_1을 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) + \begin{bmatrix} z_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = T^2\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T\begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

같은 방법으로 Z의 i번째 bit z_{i-1}를 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T^{i-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{i-2}\begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T\begin{bmatrix} z_{i-3} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_{i-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) + \begin{bmatrix} z_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= T^i\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{i-1}\begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T^2\begin{bmatrix} z_{i-3} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T\begin{bmatrix} z_{i-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

계속해서 Z의 m번째 bit z_{m-1}를 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T^{m-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-2}\begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T^{m-i}\begin{bmatrix} z_{i-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-i-1}\begin{bmatrix} z_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-i-2}\begin{bmatrix} z_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T\left(\begin{bmatrix} z_{m-3} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_{m-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) + \begin{bmatrix} z_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right)$$

$$= T^m\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-1}\begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T^{m-i+1}\begin{bmatrix} z_{i-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-i}\begin{bmatrix} z_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T^{m-i-1}\begin{bmatrix} z_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + T^2\begin{bmatrix} z_{m-3} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + T\begin{bmatrix} z_{m-2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$= S.$

이제 $\mathbb{C}$의 초기값을 모두 0으로 두고 Z'의 첫 번째 bit z_0를 입력하자. 그러면 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} z_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Z'의 두 번째 bit z_1을 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}=T^2\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_1\\0\\\vdots\\0\end{bmatrix}.$$

같은 방법으로 Z'의 i번째 bit z_{i-1}*를 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T^{i-1}\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{i-2}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\cdots+T\begin{bmatrix}z_{i-3}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{i-2}\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_{i-1}{}^*\\0\\\vdots\\0\end{bmatrix}$$

$$=T^i\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{i-1}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\cdots+T^2\begin{bmatrix}z_{i-3}\\0\\\vdots\\0\end{bmatrix}+T\begin{bmatrix}z_{i-2}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{i-1}{}^*\\0\\\vdots\\0\end{bmatrix}.$$

계속해서 Z'의 m번째 bit z_{m-1}를 입력한 후의 $\mathbb{C}$의 압축치는 아래와 같다.

$$T\left(T^{m-1}\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{m-2}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\cdots\right.$$

$$\cdots+T^{m-i}\begin{bmatrix}z_{i-2}\\0\\\vdots\\0\end{bmatrix}+T^{m-i-1}\begin{bmatrix}z_{i-1}{}^*\\0\\\vdots\\0\end{bmatrix}+T^{m-i-2}\begin{bmatrix}z_i\\0\\\vdots\\0\end{bmatrix}+\cdots$$

$$\left.\cdots+T\begin{bmatrix}z_{m-3}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}\right)+\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}$$

$$=T^m\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}+T^{m-1}\begin{bmatrix}z_0\\0\\\vdots\\0\end{bmatrix}+\cdots$$

$$\cdots+T^{m-i+1}\begin{bmatrix}z_{i-2}\\0\\\vdots\\0\end{bmatrix}+T^{m-i}\begin{bmatrix}z_{i-1}{}^*\\0\\\vdots\\0\end{bmatrix}+T^{m-i-1}\begin{bmatrix}z_i\\0\\\vdots\\0\end{bmatrix}+\cdots$$

$$\cdots+T^2\begin{bmatrix}z_{m-3}\\0\\\vdots\\0\end{bmatrix}+T\begin{bmatrix}z_{m-2}\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}z_{m-1}\\0\\\vdots\\0\end{bmatrix}$$

$$=S'.$$

이제 $S\oplus S'$를 계산해보면

$$S\oplus S'=S_e=T^{m-i}\begin{bmatrix}z_{i-1}\\0\\\vdots\\0\end{bmatrix}+T^{m-i}\begin{bmatrix}z_{i-1}{}^*\\0\\\vdots\\0\end{bmatrix}$$

$$=T^{m-i}\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$$

이고, $m-i=k$ 라 두면

$$S_e=T^k\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\ (0\le k\le m-1)$$

이다.

따라서 $i=m-k$이고, Z'의 $(m-k)$번째 bit가 오류 bit 이다.

□

$\mathbb{C}$의 상태전이 행렬 T의 특성다항식이 원시다항식이면 T에 의해 생성된 상태들이 cycle length가 2^n-1이므로 $m>2^n$인 경우 T^{-1}는 m번 clock하는 대신 $m \bmod(2^n-1)$번만 clock하면 된다.

[예제 4.2] $p(x)=x^8+x^4+x^3+x^2+1$ 원시특성다항식으로 가지는 8-cell CA를 $\mathbb{C}$라 하자. $P(x)$는 원시다항식 이다. 그러면 [20]에 의해서 $\mathbb{C}=<90,90,90,90,90,150,150,90>$임을 알 수 있다. $\mathbb{C}$의 상태전이 행렬 T는 아래와 같다.

$$T=\begin{bmatrix}01000000\\10100000\\01010000\\00101000\\00010100\\00001110\\00000111\\00000010\end{bmatrix}$$

$\mathbb{C}$의 초기값을 모두 0으로 두고 9 bit 수열 $Z=(1,0,0,1,0,1,0,0,1)$을 모두 입력시킨 후의 마지막 압축치는 $S=(1,1,0,0,1,1,0,0)$이다.

이제 Z의 7번째 bit에 오류가 있는 $Z'=(1,0,0,1,0,1,1,0,1)$에 대해서는 그림 4.1에서 보는 바와 같이 $S'=(0,1,1,0,1,1,0,0)$이다.

Z	State								Z'	State							
	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0	0	0	1	1	0	1	0	0	0
1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0	1	0	1	1	0	0	0	1	0	1
1	1	1	0	0	1	1	0	0	1	0	1	1	0	1	1	0	0
	S									S							

그림 4.1 입력수열에 따른 상태
Fig. 4.1 Input sequence and its states

$S_e = S \oplus S'$ 이므로 $S_e = (1,0,1,0,0,0,0,0)$ 이다. 그리고 T^{-1} 는 아래와 같다.

$$T^{-1} = \begin{bmatrix} 11111101 \\ 10000000 \\ 10111101 \\ 10100000 \\ 10101101 \\ 10101000 \\ 00000001 \\ 10101011 \end{bmatrix}$$

오류가 있는 bit를 찾기 위해서 $T^{-1}(S_e{}^t), T^{-2}(S_e{}^t), \cdots$ 를 순차적으로 계산하여 $T^{-k}(S_e{}^t) = [1,0,0,0,0,0,0,0]^t$ 를 만족하는 k를 찾아야 한다. 이 과정은 그림 4.2에서 보여진다. 여기에서 $S_e{}^t$는 S_e의 전치이다.

Test No.	State							
	1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	1	1	1	1	1	1	0	1
4	1	1	0	0	1	1	1	0
5	0	1	1	1	1	0	0	1
6	1	0	0	1	1	0	1	1
7	0	1	0	1	1	0	1	0
8	1	0	0	0	1	1	0	0
9	1	1	1	1	1	0	0	0

그림 4.2 상태의 역전이
Fig 4.2 Inverse transition of states

그림 4.2에서 $T^{-2}(S_e{}^t) = [1,0,0,0,0,0,0,0]^t$, $T^2[1,0,0,0,0,0,0,0]^t = S_e{}^t$이다.

이것은 $9-i=2$이고, 따라서 $i=7$이다. 결론적으로 Z'의 7번째 bit에 오류가 있음을 알 수 있다.

V. 결론 및 향후 연구방향

본 논문에서는 CA 분석기의 오류진단 방법과 오류가 있는 수열에서 오류탐지 방법을 제안하고 증명하였다. 그것을 위해 CA 분석기의 상태전이행렬과 상태전이행렬의 역행렬을 사용하였다. 테스트되는 회로에 관계없이 마지막 test 압축치가 항상 일정 하도록 초기값을 설정함으로써 CA 분석기에서 오류를 진단하였다. 이 방법에 의해 test 절차를 보다 간결하게 하였다. 그리고 CA 상태전이행렬의 역행렬을 사용하여 오류가 있는 입력수열에서 오류를 탐지하였다.

우리는 입력수열의 하나의 bit에 오류가 있는 경우에 대해서 오류탐지를 연구했다. 앞으로는 2개 이상의 bit에 오류가 있는 경우에 대해서도 연구가 이루어져야 할 것이다.

참고문헌

[1] J. von Neumann, The theory of self-reproducing automata, Z.W. Burks, Ed. Champaign, IL: Univ. Illinois Press, 1996.

[2] S. Wolfram, Statistical mechanics of cellular automata, Rev. Mod. Phys., 55, No. 3, pp. 601-644, Jul. 1983.

[3] A.K.. Das and P.P. Chaudhuri, Efficient characterization of cellular automata, in Proc. IEE(Part E) Vol. 137, No. 1, pp. 81-87, Jan. 1990.

[4] A.K. Das and P.P. Chaudhuri, Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation, IEEE Trans. Comput. Vol. 42, No. 3, pp. 340-352, Mar. 1993.

[5] P.H Bardell, Analysis of cellular automata used as pseudorandom pattern generators, in Proc. IEEE Int. test. Conf., pp. 762-767. 1990.

[6] C. Chattopadhyay, Some studies on theory and applications of additive cellular automata, Ph.D. Thesis, I.I.T., Kharagpur, India, 2002.

[7] S. Bhattacharjee, S. Shinha, C. Chattopadhyay and P.P. Chaudhuri, Cellular automata based scheme for solution of Boolean equations, in Proc. IEEE Comput. Digit. Tech. Vol. 143, No. 3, pp. 174-180, May. 1996.

[8] P. Gacs, Reliable cellular automata with self-organization, in Proc. 38th Annual Symp. Found. Comput. Sci., pp. 90-99. Oct. 1997.

[9] A. Swiedhcka and F. Seredynski, Cellular automata approach to scheduling problem, in Proc. Internat. Conf. Parallel Comput. Elect. Eng., pp. 29-33, Aug. 2000.

[10] S. Bhattacharjee, U. Raghavendra, D.R. Chowdhury and P.P. Chaudhuri, An efficient encoding algorithm for image compression hardware based on cellular automata, in Proc. 3rd Internat. Conf. High Performance Comput., pp. 239-244, Dec. 1996.

[11] K. Cattell and J.C. Muzio, Analysis of one-dimensional linear hybrid cellular automata over $GF(q)$, IEEE Trans. Comput. Vol. 45, No. 7, 782-792, Jul. 1996.

[12] S. Nandi and P.P. Chaudhuri, Analysis of periodic and intermediate boundary 90/150 cellular automata, IEEE Trans. Comput. Vol. 45, No. 1, 1-12, Jan. 1996.

[13] M. Serra, T. Slater, J.C. Muzio and D.M. Miller, The analysis of one dimensional linear cellular automata and their aliasing properties, IEEE Trans Comput. -Aided Des. Integr. Circuits Syst., Vol. 9, No. 7, pp. 767-778, Jul. 1990.

[14] S.J. Cho, U.S. Choi and H.D. Kim, Behavior of complemented CA whose complement vector is acyclic in a linear TPMACA, Math. Comput. Model., Vol. 36, No. 9/10, pp. 979-986, Dec. 2002.

[15] S.J. Cho, U.S. Choi and H.D. Kim, Analysis of complemented CA derived from a linear hybrid group CA, Comput. Math. Appl., Vol. 53, No. 1, pp. 54-63, Jan. 2007.

[16] S. Chakraborty, D.R. Chowdhury, P.P. Chaudhuri, Theory and application of nongroup cellular automata for synthesis of easily testable finite state machines, IEEE Trans. Comput., Vol. 45, No. 7, pp. 769-781, Jul. 1996.

[17] P.P. Chaudhuri, D.R. Chowdhury, S. Nandy and Chattopadhyay, Additive cellular Automata Theory and Applications, Volume 1, IEEE Computer Society Press, California, 1997.

[18] S. Nandi, B.K. Kar and P.P. Chaudhuri, Theory and application of cellular automata in cryptography, IEEE Trans. Comput., Vol. 43, No. 12, pp. 1346-1357, Dec. 1994.

[19] T. Toffoli,, Computation and Construction University of Reversible Cellular Automata, J. Comput. Syst. Sci. 15, pp. 213-231. 1977.

[20] S.J. Cho, U.S. Choi, H.D. Kim, Y.H. Hwang, J.G. Kim and S.H. Heo, New Synthesis of One-Dimensional 90/150 Linear Hybrid Group Cellular Automata, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., Vol. 26, No. 9, pp. 1720-1724, Sep. 2007.

[21] W.H. McAnney and J. Savir, Built-in Checking of the Correct Self-Test Signature, IEEE Trans. Comput., Vol. 37, No. 9, pp. 1142-1145, Sep. 1988.

[22] T.L. Booth, Sequential Machines and Automata Theory, John Willey & Sons, London, 1967.

저자소개

조성진 (Sung-Jin Cho)

1979년 강원대학교 수학교육과 학사
1981년 고려대학교 수학과 석사
1988년 고려대학교 수학과 박사
1988년 ~ 현재 부경대학교 수리과학부 정교수
※관심분야 : 셀룰라 오토마타론, 정보보호, 부호이론, 컴퓨터 구조론

권민정 (Min-Jeong Kwon)

1997년 부산대학교 수학교육과 학사
2008년 부산대학교 교육대학원 수학과 석사
2008년 ~ 현재 부경대학교 응용수학과 박사과정
※관심분야 : 셀룰라 오토마타론, 부호이론, 컴퓨터 구조론

임지미 (Ji-Mi Yim)

1997년 부산대학교 수학교육과 학사
2008년 부경대학교 교육대학원 수학과 석사
2008년 ~ 현재 부경대학교 응용수학과 박사과정
※관심분야 : 셀룰라 오토마타론, 정보보호, 부호이론

김진경 (Jin-Gyoung Kim)

2006년 부경대학교 응용수학과 석사
2008년 ~ 현재 부경대학교 응용수학과 박사과정
※관심분야 : 셀룰라 오토마타론, 유한체론, 행렬이론

박영규 (Young-Gyu Park)

2008년 ~ 현재 부경대학교 응용수학과 석사과정
※관심분야 : 셀룰라 오토마타론, 정보보호, 컴퓨터 구조론